Homework 6

Sam Fisher (A18131929)

2026-01-28

```r
library(bio3d)
```

Q6: How would you generalize the original code above to work with any set of input protein structures?

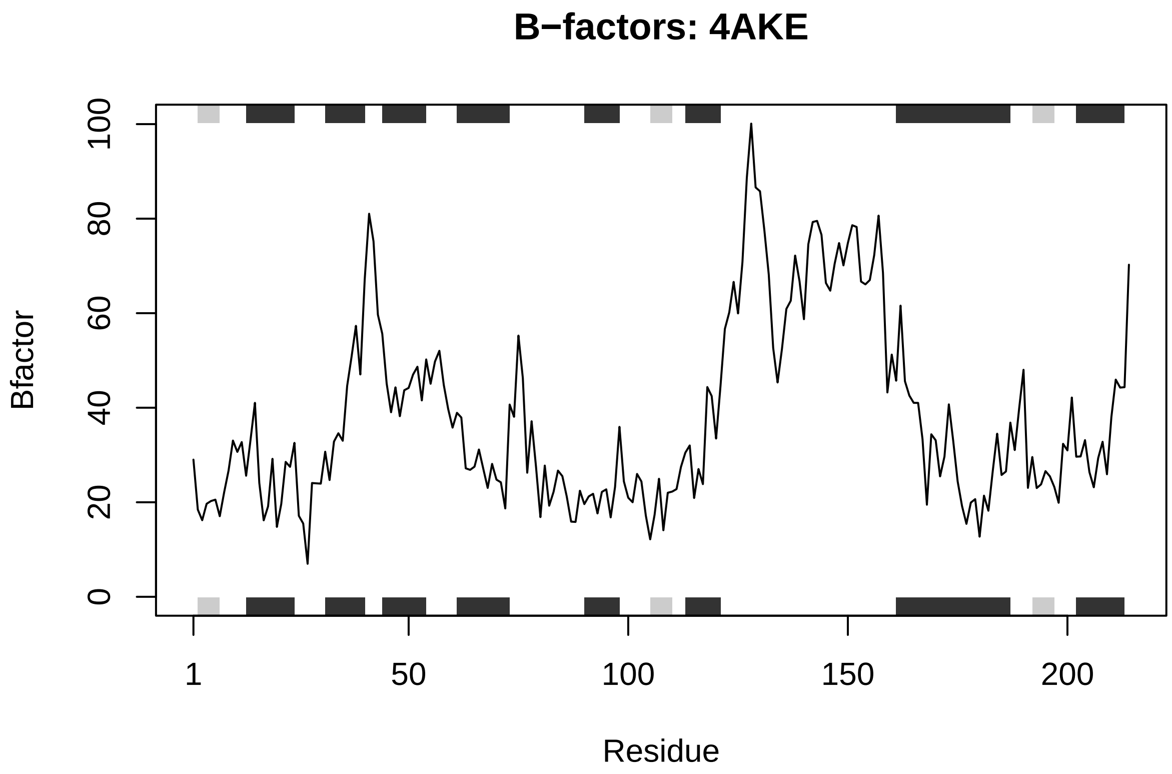
Inputs: `pdb_ids` (one or more PDB IDs), `chain` (default `"A"`), `elety` (default `"CA"`).
Purpose: Reads each PDB, cuts the selected chain and atom type, extracts B-factors, and then plots them.
Output: Returns with a named list with the trimmed PDB objects and their B-factor vectors (and produces plots).

```r
# Q6 Function: generalized B-factor analysis and plotting
# Inputs:
#   pdb_ids: character vector of PDB IDs (e.g., c("4AKE","1AKE","1E4Y")) or a single ID
#   chain: chain ID to keep (default "A")
#   elety: atom type to keep (default "CA" for C-alpha)
# Output:
#   A named list with:
#     - trimmed: trimmed pdb objects (after selecting chain/atom type)
#     - b: B-factor numeric vectors for each structure
# Side effect:
#   Produces one B-factor plot per PDB ID

analyze_bfactors <- function(pdb_ids, chain = "A", elety = "CA", typ = "l", ylab = "Bfactor") {
  # Ensure vector input
  pdb_ids <- as.character(pdb_ids)

  # Storage
  trimmed_list <- vector("list", length(pdb_ids))
  b_list <- vector("list", length(pdb_ids))
  names(trimmed_list) <- pdb_ids
  names(b_list) <- pdb_ids

  for (i in seq_along(pdb_ids)) {
    id <- pdb_ids[i]

    # Read PDB structure
    pdb <- read.pdb(id)

    # Trim to chain + atom type (e.g., chain A, C-alpha atoms)
    trimmed <- trim.pdb(pdb, chain = chain, elety = elety)

    # Extract B-factors
    b <- trimmed$atom$b
    # Plot B-factors
    plotb3(b, sse = trimmed, typ = typ, ylab = ylab, main = paste("B-factors:", id))

    # Save outputs
    trimmed_list[[id]] <- trimmed
    b_list[[id]] <- b
  }

  return(list(trimmed = trimmed_list, b = b_list))
}
```

Example usage

```
result_single <- analyze_bfactors("4AKE")
```

```
##   Note: Accessing on-line PDB file
```


B−factors: 4AKE
Bfactor
Residue